Arguments for claim 31

31. (Original) The software application of claim 30 wherein each checksum window processes 9 bytes of data 3-bytes at a time, each three-byte section treated as a single 24-bit number.

Claim 30 (and others) which serve as the basis for claim 31, were conceived and described by the applicant without awareness of the very similar Rabin-Karp algorithm. Now being aware Rabin-Karp, Claim 31 should be considered as an enhancement to that.

Background and prior art:

The invention in Claim 31 was motivated by a desire to find a faster way to scan a stream of data from a network looking for byte patterns that would indicate the presence of a computer virus in the stream. Current Anti virus products traditionally use the Aho-Corasick pattern matching algorithm. A good description of this is available on Wikipedia:

http://en.wikipedia.org/wiki/Aho-Corasick_algorithm

The fast pattern algorithm design by applicant used a modified version of hashing to achieve a substantial speed increase over Aho-Corasick.

Wikipedia describes conventional hashing here:

http://en.wikipedia.org/wiki/Hashing

Note that the Wikipedia entries cite computer virus searches as an application for Aho-Corasick, but not for hashing. Conventional hashing is not as fast for virus signature detection as Aho-Corasick, since hashing n bytes usually requires n operations.

In researching this document applicant uncovered another related algorithm, the Rabin-Karp search algorithm, described here:

http://en.wikipedia.org/wiki/Rabin-Karp_string_search_algorithm

Rabin-Karp uses the term "rolling hash" to describe what applicant referred to as "sliding window". The existence of Rabin-Karp is "prior art", and as such indicates the original "Claim 15" (sliding window) is in fact not original.

Merits of claim 31

Claim 31 introduced the idea of dividing the "sliding window" (or "rolling hash") being searched into multiple byte chunks for the purpose of computing the hash value. Specifically, claim 31 cites the example of dividing a 9 byte window into three-byte chunks. This means the hash value can be computed with single addition and subtraction operations – very fast on modern computer CPUs such as Intel Pentium and "Risk" processors such as PowerPC.

By contract, Rabin-Karp had only considered the more computationally expensive concept of treating the potion of the data stream that was in the "sliding window" (or "rolling hash") as a large based number. Implementations of Rabin-Karp thus involve multiplication operations, which are very CPU intensive.

Thus the division of the data in the sliding window allows a variation of the Rabin Karp algorithm with a substantial speed improvement, and Claim 31 should be allowed.

Respectfully Submitted,
John Alexander Bartas

Note new contract information:
John Bartas
10230 Lebanon Drive
Cupertino CA 95014
408-257-3414